11 Luhn's Test of Credit Card Numbers

In this program you will test whether a given 15 or 16-digit credit card number is valid or not. Here are the steps in your program:

- You will prompt the user to enter a 15 or 16-digit credit card number.
- You will read the credit card number in as an *int*.
- You will first check if it is a 15 or 16-digit number.
- If it is not a 15 or 16-digit number your will write an error message and exit the program using a *return* statement.
- If it is a 15 or 16-digit number you will perform the Luhn's test (outlined below).
- If the credit card number is not valid then you will print a statement to that effect.
- If the credit card number is valid then you will print the type of credit card (American Express, MasterCard, Visa, etc.) and that it is valid.

You will be making the following assumptions:

- The user enters only digits.
- The user enters digits that can be stored as a positive integer.

Here are possible scenarios that may happen. **Your output statements must be exactly the ones specified.**

Scenario 1
```
Enter 15 or 16-digit credit card number: 123456789876

Not a 15 or 16-digit number
```
Scenario 2
```
Enter 15 or 16-digit credit card number: 12345678123456780

Not a 15 or 16-digit number
```
Scenario 3
```
Enter 15 or 16-digit credit card number: 1234567891234567

Invalid credit card number
```
Scenario 4
```
Enter 15 or 16-digit credit card number: 4222222222222220

Valid Visa credit card number
```

Luhn's Test: Let us say that the credit card number was made of the following digits:

d15 d14 d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0

The last digit **d0** is the check digit in the Luhn's algorithm. The algorithm goes as follows:

- Multiply all the odd digits **d1**, **d3**, … **d15** by 2.
- Sum the digits of each product.
- Now add all the even digits **d0**, … **d14** and the single digit products of the odd digits.
- If the final sum is divisible by 10 then the credit card is valid, otherwise it is invalid.

Here are two examples worked out - one is a valid credit card number and the other is an invalid credit card number.

Valid Credit Number															
d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
4	4	3	2	3	3	3	8	8	4	1	3	8	3	4	3
8		6		6		6		16		2		16		8	
8		6		6		6		7		2		7		8	
8	4	6	2	6	3	6	8	7	4	2	3	7	3	8	3

The sum of the digits is **80**. It is divisible by 10 and hence the credit card number is valid.

Invalid Credit Number															
d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
8	2	7	3	1	2	3	2	7	3	5	1	0	5	6	9
16		14		2		6		14		10		0		12	
7		5		2		6		5		1		0		3	
7	2	5	3	2	2	6	2	5	3	1	1	0	5	3	9

The sum of the digits is **56**. It is not divisible by 10 and hence the credit card number is invalid.

To obtain the last digit of a number use the modulo (%) operator. To remove the last digit of a number divide by 10. Here is a piece of code that shows how to obtain the last two digits of the credit card number.

```
creditCard = eval (input ('Enter 15 or 16-digit credit card number:'))

d0 = creditCard % 10

creditCard = creditCard // 10

d1 = creditCard % 10
```

The first digit of a credit card is known as the Major Industry Identifier (MII).

- 1 - Airlines
- 2 - Airlines and future industry assignments
- 3 - Travel, entertainment, banking / financial
- 4 - Banking and financial
- 5 - Banking and financial
- 6 - Merchandising and banking / financial
- 7 - Petroleum and other future industry assignments
- 8 - Healthcare, telecommunications, and future industry assignments
- 9 - National assignment

The first six digits of a credit card (including the initial MII digit) are known as the issuer identification number (IIN). You are responsible for identifying only the following cards that start with the given digits:

- American Express: 34xxxx, 37xxxx
- Discover: 6011xx, 644xxx, 65xxxx

- MasterCard: 50xxxx - 55xxxx (in that range inclusive)
- Visa: 4xxxxx

If the starting digits are not the ones given, then you do not have to identify the card. The method that you will be writing will return a blank string if you cannot identify the card.

The program that you will be writing will be called CreditCard. You must use functions in your program. You can write more functions than we have asked for. At a minimum it should have the following structure:

```
  # This function checks if a credit card number is valid
  def is_valid (cc_num):

  # This function returns the type of credit card
  def cc_type (cc_num):

  def  main ():

  main()
}
```